



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,016	02/27/2004	Daryl B. Olander	ORACL-01379US0	6875
23910 7590 11/10/2008				
FLIESLER MEYER LLP				
650 CALIFORNIA STREET				
14TH FLOOR				
SAN FRANCISCO, CA 94108				
EXAMINER				
HEFFINGTON, JOHN M				
ART UNIT		PAPER NUMBER		
2179				
MAIL DATE		DELIVERY MODE		
11/10/2008		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/789,016

Applicant(s)

OLANDER ET AL.

Examiner

JOHN M. HEFFINGTON

Art Unit

2179

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 25 August 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-66 and 68 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-66 and 68 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/5508)
Paper No(s)/Mail Date 8/25/08
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date: _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

This action is in response to the amendment filed on 25 August 2008. Claims 1, 18, 34 and 50 have been amended. Claim 67 has been canceled. Claims 1-66 and 68 are pending and have been considered below.

Response to Arguments

1. Applicant's arguments filed 25 August 2008 have been fully considered but they are not persuasive. The applicant argues that a children component of an applet is not a self-contained application that renders its own GUI and the communication within one applet and anyone of its children components are internal communications within one applet. In addition, the internal communications within one applet does not resemble the communication between two portlets, each of which is a self-contained application that renders its own GUI. The examiner respectfully disagrees that this was the argument made in the rejections of the office action dated 29 May 2008. The examiner argued that the HTML page was the parent and the applets were the child components, or portlets, of the page. As presented, applets embedded in a web page can communicate with each other in a similar way as the portlets in the parent container of the instant invention.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

Art Unit: 2179

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1, 2, 4, 9-13 17, 22, 24, 34, 39, 41-45, 49, 50, 51, 53, 58, 59, 60-62 and 66 are rejected under 35 U.S.C. 102(b) as being unpatentable over Geary (Graphic Java) in view of Graham (The HTML Sourcebook Second Edition A Complete Guide to HTML 3.0).

Claim 1 and 50: Geary discloses a method of rendering a graphical user interface (GUI) comprising:

- a. providing for the representation of the GUI as a set of objects wherein the objects are organized in a logical hierarchy wherein the set of objects includes at least one of; (pages 25-26) (pages 906-917, DesktopManager) [classes and super classes, desktop management]

- b. one or more booklets wherein anyone of the one or more booklets represents a set of pages linked by a page navigator having a user selectable graphical representation and is capable of containing other booklets (pages 684-686, figures 12-15, 12-16);
- c. associating a theme with a first object in the set of objects (page 317, Pluggable Look and Feel) (page 353, Themes);
- d. rendering the first object according to the theme (page 318, figure 7-1) (page 353, Themes);
- e. rendering any descendents of the first object according to the theme (page 318, figure 7-1);
- f. wherein any descendents of the first object can override the theme (page 320, Look and Feels) [Swings pluggable look-and-feel architecture allows a component to be fitted with default properties, such as border, colors, and fonts from another component. As a result, a text area can be made to look and feel like a label.]; and
- g. wherein one of the set of objects can communicate with another of the set of objects (page 28-33, The JApplet Class) [an applet can communicate with anyone of its children components].

but does not disclose wherein the set of objects includes at least a plurality of portlets wherein anyone of the one of the plurality of portlets is a self contained application that renders its own GUI and is capable of communicating with another portlet of the plurality

of portlets. However, Graham discloses HTML code wherein one or more applets can be embedded into an HTML page. Each applet is an independent Java® application which renders its own GUI (pages 234-235). Further, official notice is taken that it is old and well known in the art that a plurality of applets embedded and executing in a single or multiple web pages can communicate with each other (Meckler, Java and Inter-Applet Communication, page 1, paragraphs 1 and 2). Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention to add the set of objects includes at least a plurality of portlets wherein anyone of the one of the plurality of portlets is a self contained application that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets to Geary. One could have been motivated to add the set of objects includes at least a plurality of portlets wherein anyone of the one of the plurality of portlets is a self contained application that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets to Geary because embedding several independent Java® Swing applications (applets) into a top level HTML page gives the user more control over multiple independent applications in a single interface. Further, as disclosed by Meckler, using a single applet in a web page to present multi-media content or simple data-entry forms works fine for small programs, but as systems get more complex, this approach becomes problematic. It would be difficult to rewrite a large desktop application as a giant monolithic applet embedded in a web page, the cost being class download time, maintenance and so on. A better approach would be a

system that supports many relatively small applets that communicate with one another, which together make up the entire application.

Claim 34: Geary discloses a method for rendering a graphical user interface (GUI), comprising:

- a. providing for the representation of the GUI as a plurality of objects wherein the objects are organized in a logical hierarchy, wherein the set of objects includes at least one of (pages 25-26) [classes and super classes] (pages 906-917, DesktopManager);
- b. one or more booklets wherein anyone of the one or more booklets represents a set of pages linked by a page navigator having a user selectable graphical representation and is capable of containing other booklets (pages 684-686, figures 12-15, 12-16);
- c. associating a first theme with a first object in the plurality of objects (page 317, Pluggable Look and Feel) (page 353, Themes);
- d. rendering the first object according to the first theme (page 318, figure 7-1) (page 353, Themes);
- e. associating a second theme with a second object in the plurality of objects (page 317, Pluggable Look and Feel) (page 353, Themes);
- f. rendering the second object according to the second theme (page 318, figure 7-1) (page 353, Themes); and

- g. wherein the second object is a descendant of the first object (page 318, figure 7-1).

but does not disclose wherein the set of objects includes at least a plurality of portlets wherein anyone of the plurality of portlets is a self contained application that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets. However, Graham discloses HTML code wherein one or more applets can be embedded into an HTML page. Each applet is an independent Java® application which renders its own GUI (pages 234-235). Further, official notice is taken that it is old and well known in the art that a plurality of applets embedded and executing in a single or multiple web pages can communicate with each other (Meckler, Java and Inter-Applet Communication, page 1, paragraphs 1 and 2). Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention to add wherein the set of objects includes at least a plurality of portlets wherein anyone of the plurality of portlets is a self contained application that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets to Geary. One could have been motivated to add wherein the set of objects includes at least a plurality of portlets wherein anyone of the plurality of portlets is a self contained application that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets to Geary because embedding several independent Java® Swing applications (applets) into a top level HTML page gives the user more control over multiple independent applications in a single interface. Further, as disclosed by Meckler, using a single applet

in a web page to present multi-media content or simple data-entry forms works fine for small programs, but as systems get more complex, this approach becomes problematic. It would be difficult to rewrite a large desktop application as a giant monolithic applet embedded in a web page, the cost being class download time, maintenance and so on. A better approach would be a system that supports many relatively small applets that communicate with one another, which together make up the entire application.

Claim 2, 39 and 51: Geary and Graham disclose the method of claim 1 wherein: one of the set of objects can respond to an event raised by another of the set of objects (page 258-262, Event Listener Lists).

Claim 4, 41 and 53: Geary and Graham disclose the method of claim 1 wherein: a control can have an interchangeable rendering mechanism (page 317, Pluggable Look and Feel) (page 353, Themes).

Claims 9, 42 and 58: Geary and Graham disclose the method of claims 1 and 34 wherein: an object can represent one of: button, text field, menu, table, window, window control, title bar, pop-up window, check-box button, radio button, window frame, desktop, shell, head, body, header, footer, book, page, layout, placeholder, portlet and toggle button (page 9, J Components).

Claim 10 and 59: Geary and Graham disclose the method of claim 1 wherein: associating the theme with the first object can occur when the first object is rendered (page 317, Pluggable Look and Feel) (page 336, figure 7-6) [Figure 7-6 shows an applet that can change look and feel upon the selection of radio buttons. Therefore, the theme is associated with the applet at render time.].

Claims 11, 43 and 60: Geary and Graham disclose the method of claims 1 and 34 wherein: the first object inherits the theme from a parent object (page 318, figure 7-1) [Figure 7-1 shows a panel with several children or descendents. These children inherit the look and feel from the parent.].

Claims 12, 44 and 61: Geary and Graham disclose the method of claims 1 and 34 wherein: the theme specifies the appearance and/or functioning of an object in the GUI (page 317, Pluggable Look and Feel).

Claims 13, 45 and 62: Geary and Graham disclose the method of claim 1 and 34 wherein: rendering the first object according to the theme can be accomplished in parallel with rendering of other objects (page 318, figure 7-1) [The parent components and child components are rendered simultaneously, i.e. in parallel.].

Claims 17, 49 and 66: Geary and Graham disclose the method of claims 1, 34 and 50 wherein: the GUI is part of a portal on the World Wide Web (page 28-33) [JApplets are a type of portal].

Claim 68: Geary and Graham disclose the method of claim 1 and Geary further discloses one of the set of objects is a desktop object and the desktop object contains one or more personalized views (pages 25-26) (pages 906-917, DesktopManager) [classes and super classes, desktop management].

5. Claims 3, 40 and 52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Geary (Graphic Java) in view of Graham (The HTML Sourcebook Second Edition A Complete Guide to HTML 3.0) and further in view of Haefel (Enterprise Java Beans).

Claim 3, 40 and 52: Geary and Graham disclose a method of rendering a GUI as in claim 1 above but do not disclose that a control or an object can have an interchangeable persistence mechanism. Haefel discloses container managed persistence in which an enterprise Java bean's (EJB) persistence is managed by the container that is using the container (page 154, Container Managed Persistence). Any container may use an EJB, therefore, an EJB container is interchangeable. Therefore, it would have been obvious to one having ordinary skill in the art to add an interchangeable persistence mechanism, since GUIs can be represented by EJBs, to Geary. One would have been motivated to use an interchangeable persistence

mechanism in order to remove the task of persistence from the responsible of the GUI programmer and to increase the flexibility of the use of the GUI, i.e. other containers may have access to the GUI.

6. Claims 5, 6, 7, 8, 18, 19-22, 24-29, 33, 35-38, 54, 55, 56 and 57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Geary in view of Graham (The HTML Sourcebook Second Edition A Complete Guide to HTML 3.0) and further in view of Schildt (Java 2, The Complete Reference).

Claims 5, 6, 7, 35-37, 54, 55 and 56: Geary and Graham disclose a method for rendering a GUI as in claims 1, 34, and 50 above but does not disclose accepting an hypertext transport protocol (HTTP) request from a web browser. Schildt discloses that a servlet can receive an HTTP from a web browser request (page 951, Life Cycle of a Servlet). Therefore, it would have been obvious to one having ordinary skill in the art to add accepting an HTTP request from a web browser to Geary. Servlets are often used to render GUIs and to receive HTTP requests from GUIs. One would have been motivated to add receiving an HTTP request from a browser to Geary in order that a JApplet may communicate with the browser that the JApplet is running in.

Claims 8, 38 and 57: Geary and Graham a method for rendering a GUI as in claims 1, 35 and 50 above, but does not disclose generating a response. Schildt discloses that a servlet can generate an HTTP response (page 951, Life Cycle of a Servlet). Servlets

are often used to generate GUIs and to send HTTP responses in response to an HTTP request. Therefore, it would have been obvious to one having ordinary skill in the art to add generating an HTTP response to Geary. One would have been motivated to add generating an HTTP response to Geary in order for the JApplet to communicate with the browser that is running the JApplet.

Claim 18: Geary discloses a method for rendering a GUI comprising:

- a. a set of objects that represent the GUI (pages 906-917, DesktopManager), wherein the set of objects includes at least one of:
 - b. one or more booklets wherein anyone of the one or more booklets represents a set of pages linked by a page navigator having a user selectable graphical representation and is capable of containing other booklets (pages 684-686, figures 12-15, 12-16);
 - c. associating a theme with a first object in the set of objects; (page 17, Pluggable Look and Feel) (page 353, Themes)
 - d. the set of objects are organized in a logical hierarchy (page 25-26)
 - e. rendering the first object according to the theme; (page 17, Pluggable Look and Feel) (page 353, Themes)
 - f. rendering any descendents of the first object according to the theme (page 318, figure 7-1); and
 - g. wherein any descendents of the first object can override the theme (page 320, Look and Feels) [Swings pluggable look-and-feel architecture allows a

component to be fitted with default properties, such as border, colors, and fonts from another component. As a result, a text area can be made to look and feel like a label.]

but does not disclose

- a. wherein the set of objects includes at least a plurality of portlets wherein anyone of the plurality of portlets is a self contained application t hat renders its own GUI that is capable of communicating with another portlet of the plurality of portlets,
- b. accepting a request, and
- c. mapping a request to a set of objects that represent the GUI

However, Graham discloses HTML code wherein one or more applets can be embedded into an HTML page. Each applet is an independent Java® application which renders it's own GUI (pages 234-235). Further, official notice is taken that it is old and well known in the art that a plurality of applets embedded and executing in a single or multiple web pages can communicate with each other (Meckler, Java and Inter-Applet Communication, page 1, paragraphs 1 and 2). Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention to add wherein the set of objects includes at least a plurality of portlets wherein anyone of the plurality of portlets is a self contained application t hat renders its own GUI that is capable of communicating with another portlet of the plurality of portlets to Geary. One could have been motivated to add wherein the set of objects includes at least a plurality of portlets

wherein anyone of the plurality of portlets is a self contained application that renders its own GUI that is capable of communicating with another portlet of the plurality of portlets to Geary because embedding several independent Java® Swing applications (applets) into a top level HTML page gives the user more control over multiple independent applications in a single interface. Further, as disclosed by Meckler, using a single applet in a web page to present multi-media content or simple data-entry forms works fine for small programs, but as systems get more complex, this approach becomes problematic. It would be difficult to rewrite a large desktop application as a giant monolithic applet embedded in a web page, the cost being class download time, maintenance and so on. A better approach would be a system that supports many relatively small applets that communicate with one another, which together make up the entire application.

Schildt discloses that a servlet can receive an HTTP request from a web browser request (page 951, Life Cycle of a Servlet). Therefore, it would have been obvious to one having ordinary skill in the art to add accepting an HTTP request from a web browser to Geary and Graham. Servlets are often used to render GUIs and to receive HTTP requests from GUIs. One would have been motivated to add receiving an HTTP request from a browser to Geary and Graham in order that a JApplet may communicate with the browser that the JApplet is running in.

Schildt also discloses that a servlet can map an HTTP request to a set of objects that represent a GUI (page 951, Life Cycle of a Servlet). It is common in the art for a servlet to render a GUI. Therefore, when a servlet receives a request and then the servlet renders a GUI, it is common for a servlet to map a request to a GUI component. Therefore, it would have been obvious to one having ordinary skill in the art to add mapping a request to a set of objects that represent the GUI to Geary and Graham. One would have been motivated to add mapping a request to a set of objects that represent the GUI to Geary and Graham so that a GUI could communicate on the World Wide Web (WWW).

Claims 19 and 20: Geary, Graham and Schildt disclose a method for rendering a GUI as in claims 18 above but does not disclose accepting an hypertext transport protocol (HTTP) request from a web browser protocol (HTTP) request. Schildt discloses that a servlet can receive an HTTP from a web browser request (page 951, Life Cycle of a Servlet). Therefore, it would have been obvious to one having ordinary skill in the art to add accepting an HTTP request from a web browser to Geary. Servlets are often used to render GUIs and to receive HTTP requests from GUIs. One would have been motivated to add receiving an HTTP request from a browser to Geary in order that a JApplet may communicate with the browser that the JApplet is running in.

Claim 21: Geary, Graham and Schildt disclose the method of claim 18, but does not disclose generating a response. Schildt discloses that a servlet can generate an HTTP

response (page 951, Life Cycle of a Servlet). Servlets are often used to generate GUIs and to send HTTP responses in response to an HTTP request. Therefore, it would have been obvious to one having ordinary skill in the art to add generating an HTTP response to Geary. One would have been motivated to add generating an HTTP response to Geary in order for the JApplet to communicate with the browser that is running the JApplet.

Claim 22: Geary, Graham and Schildt disclose the method of claim 18 and Geary further discloses one of the set of objects can respond to an event raised by another of the set of objects (page 258-262, Event Listener Lists).

Claim 24: Geary, Graham and Schildt disclose the method of claim 18 and Geary further discloses a control can have an interchangeable rendering mechanism (page 317, Pluggable Look and Feel) (page 353, Themes).

Claim 25: Geary, Graham and Shildt disclose the method of claim 18 and Geary further discloses wherein: an object can represent one of: button, text field, menu, table, window, window control, title bar, pop-up window, check-box button, radio button, window frame, desktop, shell, head, body, header, footer, book, page, layout, placeholder, portlet and toggle button (page 9, J Components).

Claim 26: Geary, Graham and Shildt disclose the method of claim 18 and Geary further discloses wherein: associating the theme with the first object can occur when the first object is rendered (page 317, Pluggable Look and Feel) (page 336, figure 7-6) [Figure 7-6 shows an applet that can change look and feel upon the selection of radio buttons. Therefore, the theme is associated with the applet at render time.].

Claim 27: Geary, Graham and Shildt disclose the method of claim 18 and Geary further discloses wherein: the first object inherits the theme from a parent object (page 318, figure 7-1) [Figure 7-1 shows a panel with several children or descendents. These children inherit the look and feel from the parent.].

Claim 28: Geary, Graham and Shildt disclose the method of claim 18 and Geary further discloses wherein: the theme specifies the appearance and/or functioning of an object in the GUI (page 317, Pluggable Look and Feel).

Claim 29: Geary, Graham and Shildt disclose the method of claim 18 and Geary further discloses wherein: rendering the first object according to the theme can be accomplished in parallel with rendering of other objects (page 318, figure 7-1) [The parent components and child components are rendered simultaneously, i.e. in parallel.].

Claim 33 Geary, Graham and Shildt disclose the method of claim 18 and Geary further discloses wherein: the GUI is part of a portal on the World Wide Web (page 28-33) [JApplets are a type of portal.].

7. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over Geary (Graphic Java) in view of Graham (The HTML Sourcebook Second Edition A Complete Guide to HTML 3.0) and Schildt (Java 2, The Complete Reference) and further in view of Haefel (Enterprise Java Beans).

Claim 23: Geary, Graham and Shildt disclose a method of rendering a GUI as in claim 18 above but do not disclose that a control or an object can have an interchangeable persistence mechanism. Haefel discloses container managed persistence in which an enterprise Java bean's (EJB) persistence is managed by the container that is using the container (page 154, Container Managed Persistence). Any container may use an EJB, therefore, an EJB container is interchangeable. Therefore, it would have been obvious to one having ordinary skill in the art to add an interchangeable persistence mechanism, since GUIs can be represented by EJBs, to Geary, Graham and Shildt. One would have been motivated to use an interchangeable persistence mechanism with Geary, Graham and Shildt in order to remove the task of persistence from the responsible of the GUI programmer and to increase the flexibility of the use of the GUI, i.e. other containers may have access to the GUI.

8. Claims 30-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Geary (Graphic Java) in view of Graham (The HTML Sourcebook Second Edition A Complete Guide to HTML 3.0) and Schildt (Java 2) as applied to claim 18 above, and further in view of Zalka (US 2004/0056894 A1).

Claim 30: Geary, Graham and Schildt disclose a method of rendering a GUI as in claim 18 above, but does not disclose wherein the theme can be specified in whole or in part by a properties file. Zalka discloses instantiating one or more extensible user interface (UI) windows based on UI window descriptions that are read and interpreted prior to automatically instating the UI windows either at or during application run time. These UI window descriptions are provided in one or more UI definition files ... (paragraph 0045). Therefore, it would have been obvious to one having ordinary skill in the art to add having UI window descriptions in UI definition files to Geary, Graham and Schildt. One would have been motivated to add having UI window descriptions in UI definition files to Geary, Graham and Schildt to make GUI properties separate from GUI functionality.

Claim 31: Geary, Graham, Schildt and Zalka disclose a method of rendering a GUI as in claim 30 above and Zalka further discloses wherein the properties file can include at least one of: 1) cascading style sheet; 2) Java Server Page; 3) Extensible Markup Language; 4) text; 5) Hypertext Markup Language; 6) Extensible Hypertext Markup Language; 7) JavaScript; and 8) Flash MX, i.e. Zalka discloses UI definitions written using eXtensible Markup Language (XML) (paragraph 0084). Therefore, it would have

been obvious to one having ordinary skill in the art to add UI definitions written using eXtensible Markup Language (XML) to Geary, Graham, Schildt and Zalka. One would have been motivated to add UI definitions written using eXtensible Markup Language (XML) to Geary, Graham, Schildt and Zalka because XML is a common language for describing UI look and feel and functionality.

Claim 32: Geary, Graham, Schildt and Zalka disclose a method of rendering a GUI as in claim 30 above but does not disclose wherein the properties file can specify at least one image. Zalka discloses UI definitions written using eXtensible Markup Language (XML). It is common in the art to specify an image in XML.

Therefore, it would have been obvious to one having ordinary skill in the art to add UI definitions written using eXtensible Markup Language (XML) (paragraph 0084) to Geary, Graham, Schildt and Zalka. One would have been motivated to add UI definitions written using eXtensible Markup Language (XML) to Geary, Graham, Schildt and Zalka because XML is a common language for describing UI look and feel and functionality.

9. Claims 14, 15, 16, 46-48, 63, 64 and 65 are rejected under 35 U.S.C. 103(a) as being unpatentable over Geary in view of Graham (The HTML Sourcebook Second Edition A Complete Guide to HTML 3.0) and further in view of Zalka (US 2004/0056894 A1).

Claims 14, 46 and 63: Geary and Graham disclose a method of rendering a GUI as in claims 1, 34 and 50 above, but do not disclose wherein the theme can be specified in whole or in part by a properties file. Zalka discloses instantiating one or more extensible user interface (UI) windows based on UI window descriptions that are read and interpreted prior to automatically instating the UI windows either at or during application run time. These UI window descriptions are provided in one or more UI definition files ... (paragraph 0045). Therefore, it would have been obvious to one having ordinary skill in the art for to add having UI window descriptions in UI definition files to Geary and Graham. One would have been motivated to add having UI window descriptions in UI definition files to Geary and Graham to make GUI properties separate from GUI functionality.

Claims 15, 47 and 64: Geary, Graham and Zalka disclose a method of rendering a GUI as in claims 14, 46 and 63 above but does not disclose wherein the properties file can include at least one of: 1) cascading style sheet; 2) Java Server Page; 3) Extensible Markup Language; 4) text; 5) Hypertext Markup Language; 6) Extensible Hypertext Markup Language; 7) JavaScript; and 8) Flash MX. Zalka discloses UI definitions written using eXtensible Markup Language (XML) (paragraph 0084). Therefore, it would have been obvious to one having ordinary skill in the art to add UI definitions written using eXtensible Markup Language (XML) to Geary, Graham and Zalka. One would have been motivated to add UI definitions written using eXtensible Markup

Language (XML) to Geary, Graham and Zalka because XML is a common language for describing UI look and feel and functionality.

Claim 16, 48 and 65: Geary, Graham and Zalka disclose a method of rendering a GUI as in claims 14, 46 and 63 above but does not disclose wherein the properties file can specify at least one image. Zalka discloses UI definitions written using eXtensible Markup Language (XML) (paragraph 0084). It is common in the art to specify an image in XML. Therefore, it would have been obvious to one having ordinary skill in the art to add UI definitions written using eXtensible Markup Language (XML) to Geary, Graham and Zalka. One would have been motivated to add UI definitions written using eXtensible Markup Language (XML) to Geary, Graham and Zalka because XML is a common language for describing UI look and feel and functionality.

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John M. Heffington whose telephone number is (571) 270-1696. The examiner can normally be reached on Mon - Fri 8:00 - 5:30 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JMH
11/7/08

/Ba Huynh/

Application/Control Number: 10/789,016

Page 24

Art Unit: 2179

Primary Examiner, Art Unit 2179